CE343 SOFTWARE ENGINEERING

Chapter 4 Requirement Analysis & Specification

Prepared by:

Ms. Sneha Padhiar

Assistant Professor,

U & P U. Patel Department of Computer Engineering

Requirements Analysis and Specification

- Many projects fail:
  - Because they start implementing the system.
  - Without determining whether they are building what the customer really wants.

Requirements Analysis and Specification(cont..)

- Goals of requirements analysis and specification phase:
  - Fully understand the user requirements.
  - Remove inconsistencies, anomalies,etc from requirements.
  - Document requirements properly in an SRS document.

Requirements Analysis and Specification(cont..)

- Consists of two distinct activities:
  - Requirements Gathering and Analysis
  - Specification

Who Carries Out Requirements Analysis and Specification?

- The person who undertakes requirements analysis and specification:
  - Known as systems analyst:
  - Collects data pertaining to the product
  - Analyzes collected data:
    - To understand what exactly needs to be done.
  - Writes the Software Requirements Specification (SRS) document.

Requirements Gathering Activities

1. Studying the existing documentation
2. Interview
3. Task analysis
4. Scenario analysis
5. Form analysis

Analysis of the Gathered Requirements

- Main purpose of requirements analysis:
  - Clearly understand the user requirements,
  - Detect inconsistencies, ambiguities, and incompleteness.
- Incompleteness and inconsistencies:
  - Resolved through further discussions with the end-users and the customers.

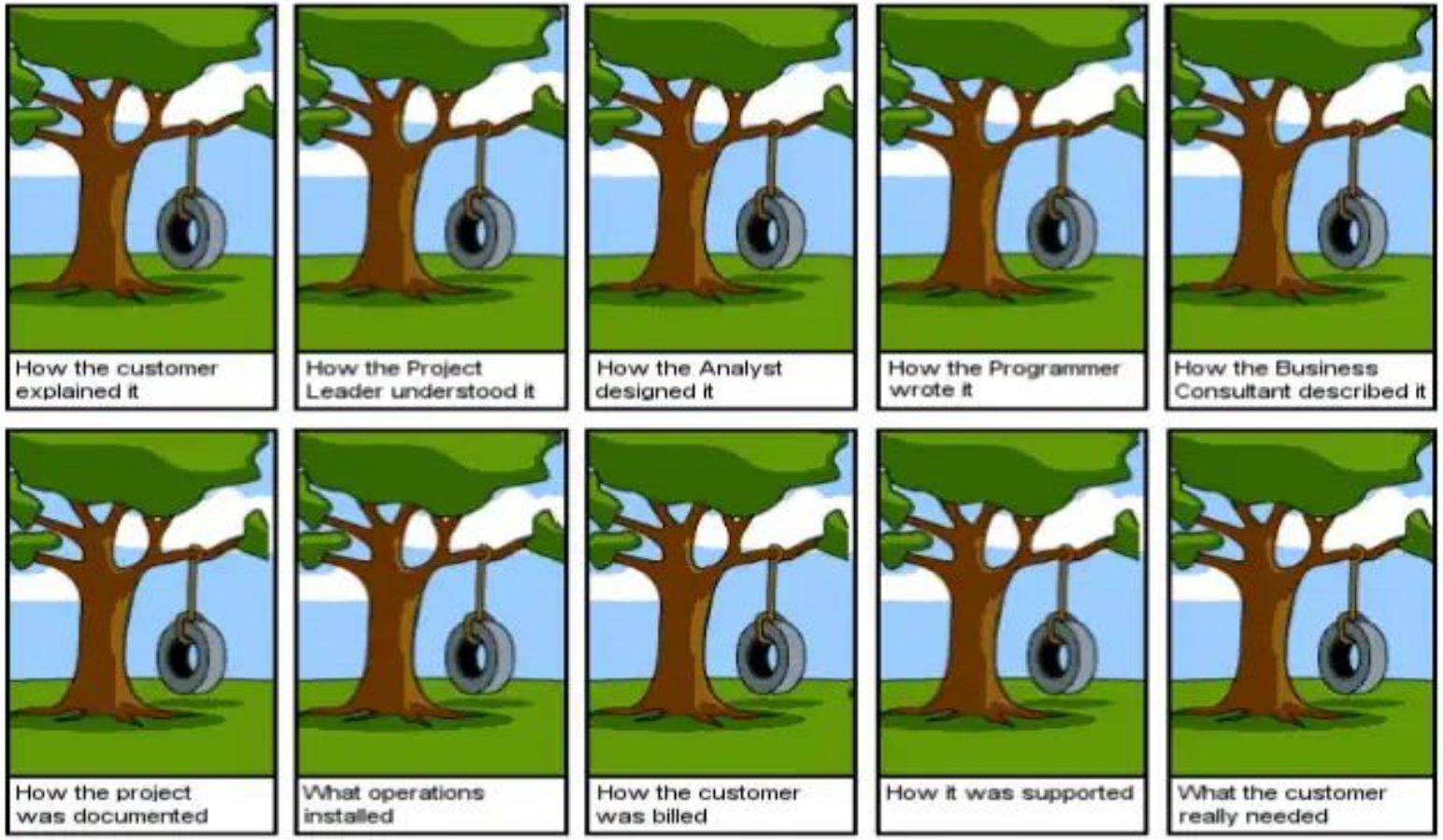
Inconsistent Requirement

- Some part of the requirement:
  - contradicts with some other part.
- Example:
  - One customer says turn off heater and open water shower when temperature > 100 C
  - Another customer says turn off heater and turn ON cooler when temperature > 100 C

Incomplete Requirement

- Some requirements have been omitted:
  - Possibly due to oversight.
- Example:
  - The analyst has not recorded:
    when temperature falls below 90 C
    - heater should be turned ON
    - water shower turned OFF.

Ideal Case-tree swing engineering joke

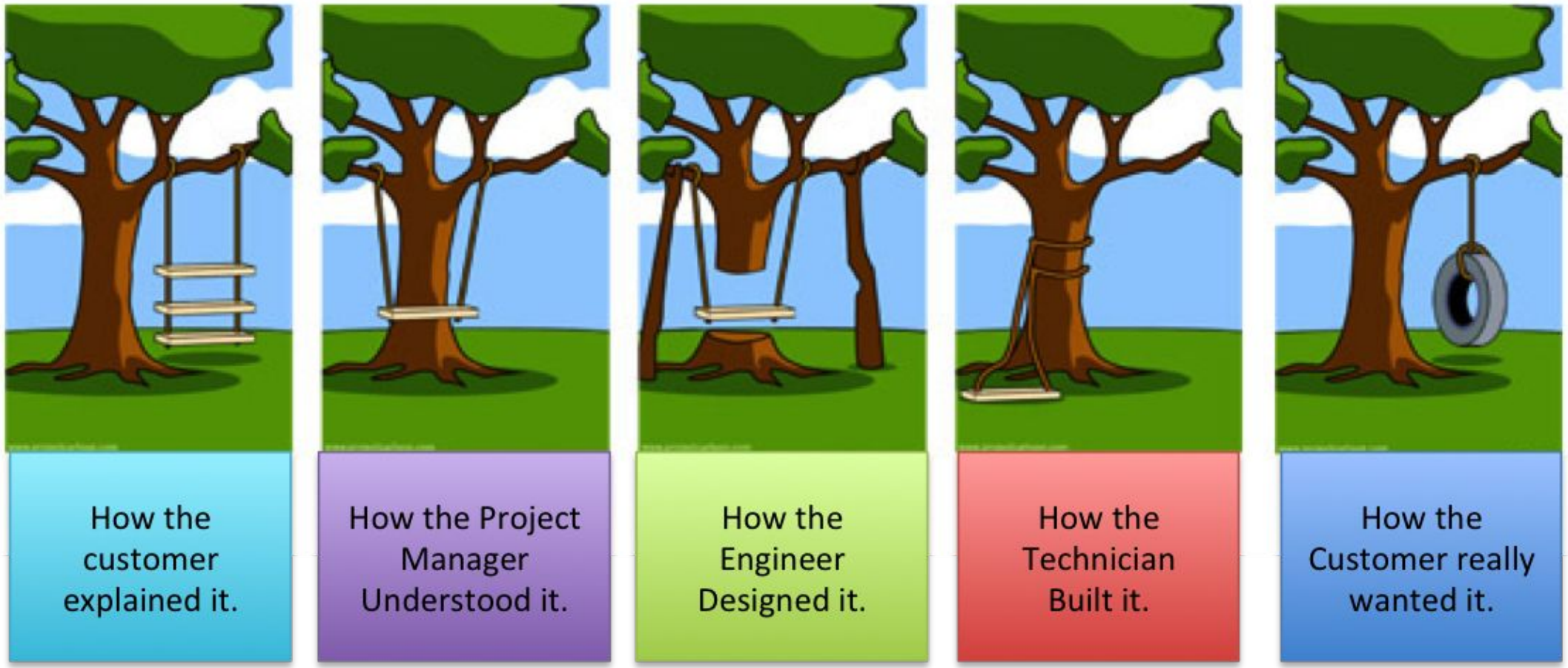


D. Monett – Europe Week 2014, University of Hertfordshire, Hatfield

Adapted from cartoon © http://projectcartoon.com/

67

Actual Case-tree swing engineering joke



Analysis of the Gathered Requirements(CONT.)

- Several things about the project should be clearly understood by the analyst:
  - What is the problem?
  - Why is it important to solve the problem?
  - What are the possible solutions to the problem?
  - What complexities might arise while solving the problem?

Software Requirements Specification

- Main aim of requirements specification:
  - Systematically organize the requirements arrived during requirements analysis.
  - Document requirements properly.

Software Requirements Specification

- The SRS document is useful in various contexts:
  - Statement of user needs
  - Contract document
  - Reference document
  - Definition for implementation

Software Requirements Specification: A Contract Document

- Requirements document is a reference document.
- SRS document is a contract between the development team and the customer.
  - Once the SRS document is approved by the customer,
    - Any subsequent controversies are settled by referring the SRS document.

Software Requirements Specification: A Contract Document

- Once customer agrees to the SRS document:
  - Development team starts to develop the product according to the requirements recorded in the SRS document.
- The final product will be acceptable to the customer:
  - As long as it satisfies all the requirements recorded in the SRS document.

SRS Document (CONT.)

- The SRS document is known as black-box specification:
  - The system is considered as a black box whose internal details are not known.
  - Only its visible external (i.e. input/output) behaviour is documented.

SRS Document (CONT.)

- SRS document concentrates on:
  - What needs to be done
  - Carefully avoids the solution ("how to do") aspects.
- The SRS document serves as a contract
  - Between development team and the customer.
  - Should be carefully written

Properties of a Good SRS Document

- It should be concise
  - and at the same time should not be ambiguous.
- It should specify what the system must do
  - and not say how to do it.
- Easy to change.,
  - i.e. it should be well-structured.
- It should be consistent.
- It should be complete.

Properties of a Good SRS Document (cont...)

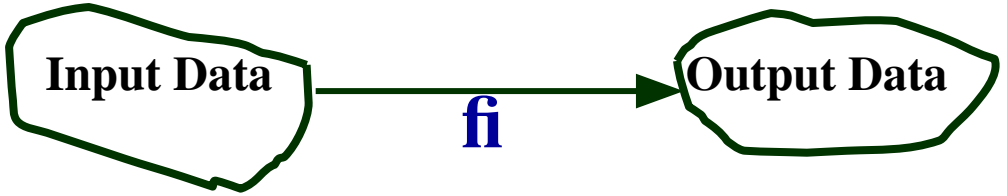
- It should be traceable
  - You should be able to trace which part of the specification corresponds to which part of the design, code, etc and vice versa.
- It should be verifiable
  - e.g. “system should be user friendly” is not verifiable

SRS Document (CONT.)

- SRS document, normally contains three important parts:
  - Functional requirements,
  - Non-functional requirements,
  - Goals of Implementation.

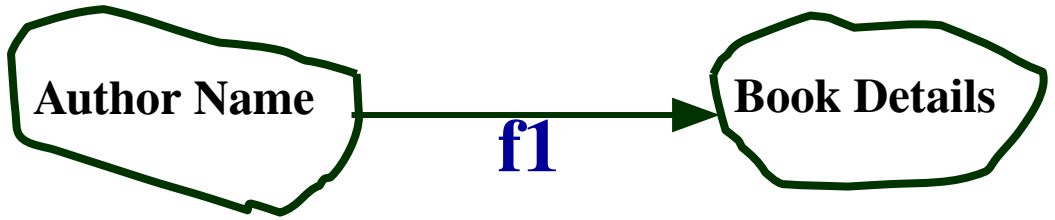
SRS Document (CONT.)

- It is desirable to consider every system:
  - Performing a set of functions {fi}.
  - Each function fi considered as:
  - Transforming a set of input data to corresponding output data.



Example: Functional Requirement

- F1: Search Book
  - Input:
    - an author's name:
  - Output:
    - details of the author's books and the locations of these books in the library.



Functional Requirements

- Functional requirements describe:
  - A set of high-level requirements
  - Each high-level requirement:
    - takes in some data from the user
    - outputs some data to the user
  - Each high-level requirement:
    - might consist of a set of identifiable functions

Functional Requirements

- For each high-level requirement:
  - Every function is described in terms of:
    - Input data set
    - Output data set
    - Processing required to obtain the output data set from the input data set.

Example Functional Requirements

- Req. 1:
  - Once the user selects the “search” option,
    - he is asked to enter the key words.
  - The system should output details of all books
    - whose title or author name matches any of the key words entered.
    - Details include: Title, Author Name, Publisher name, Year of Publication, ISBN Number, Catalog Number, Location in the Library.

R.1.1:

Input: “search” option,
Output: user prompted to enter the key words.

R1.2:

Input: key words
Output: Details of all books whose title or author name matches any of the key words.
Details include: Title, Author Name, Publisher name, Year of Publication, ISBN Number, Catalog Number, Location in the Library.
Processing: Search the book list for the keywords

Example Functional Requirements

- Req. 2:
  - When the “renew” option is selected,
    - The user is asked to enter his membership number and password.
  - After password validation,
    - The list of the books borrowed by him are displayed.
  - The user can renew any of the books:
    - By clicking in the corresponding renew box.

R2.1:

Input: “renew” option selected,
Output: user prompted to enter his membership number and password.

R2.2:

Input: membership number and password
Output:
list of the books borrowed by user are displayed. User prompted to enter books to be renewed or
user informed about bad password
Processing: Password validation, search books issued to the user from borrower list and display.

R2.3:

Input: user choice for renewal of the books issued to him through mouse clicks in the corresponding renew box.

Output: Confirmation of the books renewed

Processing: Renew the books selected by the in the borrower list.

Nonfunctional Requirements

- Characteristics of the system which can not be expressed as functions:
  - Maintainability,
  - Portability,
  - Usability, etc.

Nonfunctional Requirements

- Nonfunctional requirements include:
  - Reliability issues,
  - Performance issues:
    - Example: How fast the system can produce results
      - so that it does not overload another system to which it supplies data, etc.
  - Human-computer interface issues,
  - Interface with other external systems,
  - Security, maintainability, etc.

Non-Functional Requirements

- Hardware to be used,
- Operating system
  - or DBMS to be used
- Capabilities of I/O devices
- Standards compliance
- Data representations
  - by the interfaced system

Organization of the SRS Document

- Introduction.
- Functional Requirements
- Non-functional Requirements
  - External interface requirements
  - Performance requirements
- Goals of implementation

DIFFERENCE

Sl.No	Functional Requirement	Non-functional Requirement
1.	Defines all the services or functions required by the customer that must be provided by the system	Defines system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
2.	It describes what the software should do.	It does not describe what the software will do, but how the software will do it.
3.	Related to business. For example: Calculation of order value by Sales Department or gross pay by the Payroll Department	Related to improving the performance of the business. For example: checking the level of security. An operator should be allowed to view only my name and personal identification code.
4.	Functional requirement are easy to test.	Nonfunctional requirements are difficult to test
5.	Related to the individual system features	Related to the system as a whole
6.	Failure to meet the individual functional requirement may degrade the system	Failure to meet a non-functional requirement may make the whole system unusable.

Examples of Bad SRS Documents

- Unstructured Specifications:
  - Narrative essay --- one of the worst types of specification document:
    - Difficult to change,
    - Difficult to be precise,
    - Difficult to be unambiguous,
    - Scope for contradictions, etc.

Examples of Bad SRS Documents

- Noise:
  - Presence of text containing information irrelevant to the problem.
- Silence:
  - aspects important to proper solution of the problem are omitted.

Examples of Bad SRS Documents

- Over specification:
  - Addressing “how to” aspects
  - For example, “Library member names should be stored in a sorted descending order”
  - Over specification restricts the solution space for the designer.
- Contradictions:
  - Contradictions might arise
    - if the same thing described at several places in different ways.

Examples of Bad SRS Documents

- Ambiguity:
  - Literary expressions
  - Unquantifiable aspects, e.g. “good user interface”
- Forward References:
  - References to aspects of problem
    - defined only later on in the text.
- Wishful Thinking:
  - Descriptions of aspects
    - for which realistic solutions will be hard to find.

Requirement Engineering

- ✔ Requirements inception or requirements elicitation(Gathering)
- ✔ Requirements analysis and negotiation
- ✔ System modeling
- ✔ Requirements specification
- ✔ Requirements validation
- ✔ Requirements management